

T S2/5/1

2/5/1

DIALOG(R)File 351:Derwent WPI

(c) 2004 Thomson Derwent. All rts. reserv.

010038893 \*\*Image available\*\*

WPI Acc No: 1994-306604/199438

XRPX Acc No: N94-241218

**Signal encoding and decoding device - resets decimal point position at optimum value if operation result of conversion process overflows**

Patent Assignee: SANYO ELECTRIC CO LTD (SAOL )

Number of Countries: 001 Number of Patents: 002

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
JP 6232762	A	19940819	JP 9316392	A	19930203	199438 B
JP 3138100	B2	20010226	JP 9316392	A	19930203	200114

Priority Applications (No Type Date): JP 9316392 A 19930203

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
JP 6232762	A	13	H03M-007/30	
JP 3138100	B2	14	H03M-007/30	Previous Publ. patent JP 6232762

Abstract (Basic): JP 6232762 A

The signal encoder has a filter (3) for block changing of the input signal and dividing it for each predetermined frequency band. The block changed input signals are started on a time-base using a waveform logging circuit (2). The input signals on the time base are changed into signals on a frequency axis by a rectifying calculation device (4). The decimal point position for each group of signals is usually set-up arbitrarily.

If the result of the conversion process overflows, the decimal point position is reset at an optimum value. The output signals of the rectifying calculation device are quantised using a quantisation device (5). The decoder performs all reverse operations accordingly so that decoding of the frequency-based output signals of the encoder is carried out to output the original input signal of the time-base.

ADVANTAGE - Raises accuracy without increasing circuit scale. Provides restoring of overflowing data into exact data.

Dwg.1/3

Title Terms: SIGNAL; ENCODE; DECODE; DEVICE; RESET; DECIMAL; POINT; POSITION; OPTIMUM; VALUE; OPERATE; RESULT; CONVERT; PROCESS; OVERFLOW

Derwent Class: P86; T01; U21; W04

International Patent Class (Main): H03M-007/30

International Patent Class (Additional): G06F-007/38; G06F-015/332; G06F-017/14; G10L-007/04; G10L-009/16; G10L-011/00; G10L-019/02

File Segment: EPI; EngPI

?

***This Page Blank (uspto)***



JPA 6-232762

## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **06232762 A**(43) Date of publication of application: **19.08.94**

(51) Int. Cl. **H03M 7/30**  
**G06F 7/38**  
**G06F 15/332**  
**G10L 7/04**  
**G10L 9/16**

(21) Application number: **05016392**(22) Date of filing: **03.02.93**(71) Applicant: **SANYO ELECTRIC CO LTD**(72) Inventor: **FUMA MASATO**  
**OKAMOTO SANEYUKI**(54) **SIGNAL CODER AND SIGNAL DECODER**

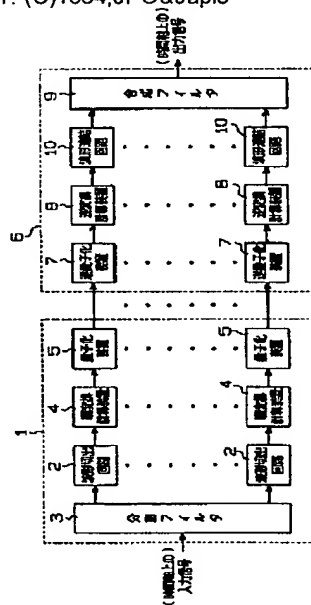
signal coder 1.

## (57) Abstract:

COPYRIGHT: (C)1994,JPO&amp;Japio

**PURPOSE:** To obtain a signal coder and the signal decoder in which the accuracy is enhanced without increasing its circuit scale.

**CONSTITUTION:** A division filter 3 divides an input signal into blocks for each prescribed frequency band. A waveform segmentation circuit 2 segments the input signal on a time axis for each prescribed time through block processing. Each forward conversion calculation device 4 converts an input signal on the time axis into a signal on a frequency axis. In this case, the fixed decimal point system is adopted for each group, the decimal point position is set optionally by each group and when the result of arithmetic operation for the conversion processing indicates overflow, the position of the decimal point is set again to an optimum value. Each quantization device 5 quantizes an output signal of each forward conversion calculation device 4 and outputs the result. A signal decoder 6 applies inverse processing to an output signal on the frequency axis of the signal coder 1 and decodes the signal into the original input signal on the time axis inputted to the



**This Page Blank (uspto)**

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平6-232762

(43) 公開日 平成 6 年(1994) 8 月19日

(51) Int. Cl. <sup>5</sup>	識別記号	庁内整理番号	F I	技術表示箇所
H03M 7/30		A 8522-5J		
G06F 7/38		Y 9291-5B		
15/332		S 7343-5L		
G10L 7/04		G 8946-5H		
9/16		8946-5H		

審査請求 未請求 請求項の数 3 O L (全13頁)

(21) 出願番号 特願平5-16392

(22) 出願日 平成 5 年(1993) 2 月 3 日

(71) 出願人 000001889

三洋電機株式会社

大阪府守口市京阪本通 2 丁目 5 番 5 号

(72) 発明者 夫馬 正人

大阪府守口市京阪本通 2 丁目18番地 三洋  
電機株式会社内

(72) 発明者 岡本 実幸

大阪府守口市京阪本通 2 丁目18番地 三洋  
電機株式会社内

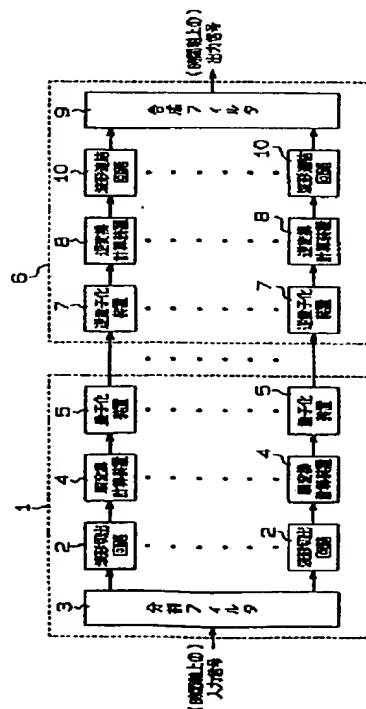
(74) 代理人 弁理士 西野 卓嗣

(54) 【発明の名称】 信号符号化装置および信号復号化装置

(57) 【要約】

【目的】 回路規模を増大させることなく精度を高めることが可能な信号符号化装置および信号復号化装置を提供する。

【構成】 分割フィルタ 3 は入力信号を所定の周波数帯域毎にブロック化して分割する。波形切り出し回路 2 は時間軸上の入力信号を所定の時間毎にブロック化して切り出す。各順変換計算装置 4 は時間軸上の入力信号を周波数軸上に変換する。その際に、各グループ毎に固定小数点方式をとり、その小数点位置を各グループ毎に任意に設定すると共に、変換のための演算処理において演算結果がオーバーフローした場合には、小数点位置を最適な値に設定し直す。各量子化装置 5 は各順変換計算装置 4 の出力信号を量子化して出力する。信号復号化装置 6 は、信号符号化装置 1 の周波数軸上の出力信号に信号符号化装置 1 とは逆の処理を施して、信号符号化装置 1 に入力された元の時間軸上の入力信号に復号化する。



## 【特許請求の範囲】

【請求項 1】 入力信号波形を所定の周波数帯域毎にブロック化して分割すると共に、そのブロック内の信号を所定の時間毎にブロック化して切り出し、その分割されて切り出された各グループの信号を相互に独立な変換軸で変換して符号化する際に、各グループ毎に固定小数点方式をとり、その小数点位置を各グループ毎に任意に設定すると共に、変換のための演算処理において演算結果がオーバーフローした場合には、小数点位置を最適な値に設定し直すことを特徴とする信号符号化装置。

【請求項 2】 請求項 1 記載の信号符号化装置から出力される各グループ毎の信号を相互に独立な変換軸で変換して復号化する際に、各グループ毎に固定小数点方式をとり、その小数点位置を各グループ毎に任意に設定すると共に、変換のための演算処理において演算結果がオーバーフローした場合には、小数点位置を最適な値に設定し直すことを特徴とする信号復号化装置。

【請求項 3】 請求項 1 記載の信号符号化装置または請求項 2 記載の信号復号化装置において、前記各グループの信号をデータとして処理する際に、データのサインビットに適宜なビット数のマージンビットを付加して複数ビットから成るサインビットとし、前記演算処理は当該マージンビットのビット数を越えないように設定して、前記オーバーフローは当該マージンビットの演算処理による変化から判定すると共に、そのマージンビットの変化からオーバーフローしたデータを正確なデータに復元することを特徴とする信号符号化装置または信号復号化装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 本発明は信号符号化装置および信号復号化装置に関するものである。

## 【0002】

【従来の技術】 近年、信号圧縮符号化の一つとして変換符号化が多用されている。変換符号化では、入力信号を相互に独立な変換軸、例えば、直交する変換軸で変換（すなわち、直交変換）した後に量子化して出力する。この変換符号化された出力信号を元の入力信号に復号化するには、前記符号化と逆の処理、つまり、逆量子化した後に逆直交変換を施せばよい。

【0003】 ところで、直交変換には、アダマール変換、カルーネンレーブ変換、ルジャンドル変換、離散的フーリエ変換（DFT ; Discrete Fourier Transform）、離散的余弦変換（DCT ; Discrete Cosine Transform）等がある。

【0004】 この中で、DCTについては、オーディオのデジタル信号を扱うために特に改良を加えた、改良離散的余弦変換（MDCT ; Modified Discrete Cosine Transform）がある。このMDCTでは時間軸上の信号を周波数軸上に変換することができ、MDCTの線形逆

変換であるIMDCT（Inverse Modified Discrete Cosine Transform）では周波数軸上の信号を時間軸上に変換することができる。尚、MDCTについては、アイイーイーイー・トランザクションズ・オン・エイエスエスピー（IEEE TRANSACTIONS ON ASSP）34巻5号、1986年、1153～1161頁に詳述されている。

【0005】 ところで、上記文献のMDCTまたはIMDCTでは、線形順変換または線形逆変換にて積和演算を用いている。そのため、ブロック長（サンプル数）Nが大きくなると乗算回数および加算回数が、ほぼNの2乗に比例して増加してしまい、処理に時間がかかりすぎるという問題があった。

【0006】 そこで、積和演算で行っていた線形順変換および線形逆変換に、高速フーリエ変換（FFT ; Fast Fourier Transform）を利用して処理時間を短縮する方法が、特開平4-44099号公報に開示されている。

【0007】 この方法において、線形順変換を行う順変換計算装置は、入力バッファと順変換用ウィンドウ部と線形順変換部とから構成されている。入力バッファはNサンプルの入力信号を保持する。順変換用ウィンドウ部は、入力バッファの出力信号にそれぞれ順変換用ウィンドウ関数を乗算する。線形順変換部は前処理用乗算部とFFT部と後処理用乗算部とから構成され、順変換用ウィンドウ部の出力信号に線形順変換を施す。

【0008】 すなわち、前処理用乗算部は、順変換用ウィンドウ部で順変換用ウィンドウ関数を乗算して得られたNサンプル（第0番～第N-1番まで）の入力信号に対して、第0番から第N/4-1番までのN/4サンプルの入力信号を逆極性として第3N/4番から第N番までの中間信号とする。また、前処理用乗算部は、第N/4番から第N-1番までの3N/4サンプルの入力信号を第0番から第3N/4-1番までの中間信号とする。そして、前処理用乗算部は、得られたNサンプルの中間信号に $\exp(-2\pi j n/N)$ を乗算する。

【0009】 FFT部は、前処理用乗算部の出力信号にFFTを施して時間軸上から周波数軸上に変換する。後処理用乗算部は、FFT部のNサンプルの出力信号に $\exp(-2\pi j (k+1/2)/2N)$ を乗算して実数成分を出力する。

【0010】 一方、線形逆変換を行う逆変換計算装置は、線形逆変換部と逆変換用ウィンドウ部と出力バッファとから構成されている。線形逆変換部は前処理用乗算部とIFFT部と後処理用乗算部とから構成され、入力信号（すなわち、順変換計算装置の出力信号）に線形逆変換を施す。逆変換用ウィンドウ部は、線形逆変換部の出力信号にそれぞれ逆変換用ウィンドウ関数を乗算する。出力バッファは逆変換用ウィンドウ部の出力信号を保持する。

【0011】 ここで、前処理用乗算部は、順変換計算装置によって生成されたNサンプルの信号に $\exp\{2\pi$

$j(N/4 + 1/2)k/N$  を乗算する。IFFT部は、前処理用乗算部の出力信号に逆高速フーリエ変換 (IFFT; Inverse Fast Fourier Transform) を施して周波数軸上から時間軸上に変換する。

【0012】後処理用乗算部は、IFFT部のNサンプルの出力信号に  $\exp\{2\pi j(n + N/4 + 1/2)/2N\}$  を乗算して実数成分を出力する。但し、 $n$  と  $k$  は0からN-1までの整数である。

【0013】ここで、順変換計算装置の各部において必要な乗算回数は、前処理乗算部においてN回、FFT部で最大  $N\log_2 N$  回、後処理乗算部でN回であり、順変換計算装置全体で必要な乗算回数の合計  $N(2 + \log_2 N)$  はNが大きくなると、ほぼ  $N\log_2 N$  回に等しくなる。また、順変換計算装置において必要な加算回数は、FFT部における  $2N\log_2 N$  回である。

【0014】一方、逆変換計算装置の各部において必要な乗算回数は、前処理乗算部においてN回、IFFT部で最大  $N\log_2 N$  回、後処理乗算部でN回であり、逆変換計算装置全体で必要な乗算回数の合計  $N(2 + \log_2 N)$  はNが大きくなると、順変換計算装置と同様にほぼ  $N\log_2 N$  回に等しくなる。また、逆変換計算装置において必要な加算回数は、IFFT部における  $2N\log_2 N$  回である。

【0015】従って、この方法では線形順変換および線形逆変換に必要な乗算回数および加算回数がほぼ  $N\log_2 N$  に比例して増加するため、ほぼNの2乗に比例して増加する前記方法に比べて、処理時間を大幅に短縮することができる。

【0016】

【発明が解決しようとする課題】ところで、上記の順変換計算装置および逆変換計算装置を具体化する場合は、前記入力バッファまたは出力バッファとして、装置内にワークRAMを設けるのが一般的である。そして、演算処理に必要なデータ（前記入力信号、中間信号、出力信号、演算途中の信号等）をそのワークRAMから読み出して演算し、演算結果を再びワークRAMに書き込むという処理を繰り返し行うようになっている。また、データのワークRAMへの格納方法（すなわち、表示方法）および演算方法においては、固定小数点方式または浮動小数点方式をとっている。

【0017】図8に、固定小数点方式をとる順変換計算装置51のブロック回路図を示す。順変換計算装置51は、固定小数点演算を行うMDCT演算回路53とRAM52とから構成されている。

【0018】前記入力バッファとしてのRAM52は、実時間上のサンプル点がN個（つまり、時間軸上のN点の離散値によって信号が表される）である入力信号を入力して保持する。

【0019】前記順変換ウィンドウ部および線形順変換部としてのMDCT演算回路53は、RAM52から所

望の入力信号を読み出しては前記した演算処理を行い、その結果を再びRAM52に書き込むという動作を繰り返し行う。

【0020】これにより、Nサンプルの入力信号の全てについて所定の演算処理が終了すると、RAM52に保持されていたNサンプルの入力信号は、入力信号を周波数軸上のN点の離散値に変換した信号であるNサンプルの出力信号に置き代わることになる。

【0021】ここで、RAM52中のデータの表示方法は固定小数点表示であるため、例えば、各データのデータ長を24ビットに設定した場合、RAM52内における1つ1つのデータの格納状態は図10に示すようになる。すなわち、最上位ビット（24ビット目）をサインビットとして、残りの23ビットで1つのデータが表される。このとき、小数点は各データの大きさ（すなわち、信号のレベルの大小）に関係なく、予め定められた位置に固定されている（図10においては、7ビットと8ビットの間）。

【0022】ところで、MDCT演算回路53内における演算処理には誤差がつきものであり、その誤差がノイズとなって信号に影響を与える。従って、固定小数点方式によれば、大きなレベルの信号について十分なS/N比を得るように小数点位置を設定すると、小さなレベルの信号についてはS/N比が悪化してしまう。

【0023】図12および図13は、Nサンプルの全データ（すなわち、全信号）のデータ長を24ビットに設定し、小数点位置を7ビットと8ビットの間に固定した場合における、RAM52内の全信号の格納状態を模式的に示したものである。

【0024】図12に示すように、信号のレベルが全体的に大きくなるときはS/N比も大きくなる。一方、図13に示すように、信号のレベルが全体的に小さくなるときはS/N比も小さくなってしまい、ノイズレベルに埋もれることになってS/N比が悪化してしまう。

【0025】ここで、小数点を7ビットと8ビットの間ではなく、もっと高いビット（例えば、十数ビット）に固定すれば、図13に示すような全体的に小さな信号に対して十分なS/N比を得ることができる。しかしながら、その場合は、図12に示すような大きな信号に対してオーバーフローしてしまうことになる。すなわち、固定小数点方式によれば、小さなレベルの信号について十分なS/N比を得るように小数点位置を設定すると、大きなレベルの信号についてはオーバーフローして歪んでしまう。

【0026】このように、固定小数点方式をとる順変換計算装置51においては、大きなレベルの信号に合わせて小数点位置を固定すると、小さなレベルの信号のS/N比が悪化し、反対に、小さなレベルの信号に合わせて小数点位置を固定すると、大きなレベルの信号が歪んでしまうという問題があった。すなわち、レベル差が大き

な信号を扱う場合には十分なS/N比がとれず、精度が低くなるという問題があった。

【0027】また、MDCT演算回路53による演算処理においては、演算処理を繰り返す度にデータの値が大きくなることもある。そのため、演算処理によってデータがオーバーフローし、その結果がRAM52に書き込まれると、次に、そのオーバーフローしたデータをRAM52から読み出して演算処理した結果も同様にオーバーフローしてしまう。すなわち、演算処理によってデータが一旦オーバーフローすると、それ以降の演算結果は全てオーバーフローすることになり意味をなさなくなる。

【0028】そこで、固定小数点方式をとる順変換計算装置51においては、小数点位置をできるだけ低いビットに固定して、Nサンプルの全データについて所定の演算処理が終了するまで絶対にオーバーフローしないようにする必要がある。しかしながら、小数点位置を低いビットに固定すると、前記したように、大きなレベルの信号については十分なS/N比が得られる反面、小さなレベルの信号のS/N比が悪化してしまうという問題があった。

【0029】図9に、浮動小数点方式をとる順変換計算装置61のブロック回路図を示す。順変換計算装置61は、MDCT演算回路63とRAM62と小数点位置レジスタ64とから構成されている。

【0030】ここで、浮動小数点方式の順変換計算装置61と固定小数点方式の順変換計算装置51との違いは下記の点についてだけであり、全体の動作は同じである。

①MDCT演算回路63が固定小数点演算ではなく浮動小数点演算を行う。

【0031】②RAM62内の各データ毎に小数点位置レジスタ64が設けられており、各データは浮動小数点表示される。

上記②については、例えば、各データのデータ長を24ビットに設定した場合、RAM62内における1つ1つのデータの格納状態は図11に示すようになる。すなわち、最上位ビット(24ビット目)をサインビットとして、残りの23ビットで1つのデータが表される。このとき、当該データの小数点位置は、各データ毎に設けられた5ビットの小数点位置レジスタ64に格納されている。そのため、各データの小数点位置は任意に設定することができる。

【0032】従って、浮動小数点方式によれば信号のレベルの大小に応じて最適な小数点位置を設定することができ、固定小数点方式における上記問題点を解決することができる。

【0033】しかしながら、浮動小数点方式の順変換計算装置61においては、小数点位置レジスタ64を各データ毎に設けなければならないため、サンプル数Nが多

くなると全体の回路規模が著しく大きくなるという問題があった。

【0034】さらに、MDCT演算回路63において、小数点位置が異なるデータ同志を演算処理する際には、小数点位置を合わせる操作(いわゆる、スケール合わせ)が必要であるため、処理が複雑になって回路規模が大きくなるという問題もあった。

【0035】また、逆変換計算装置と順変換計算装置との違いは、上記MDCT演算回路53、63を前記線形逆変換部および逆変換ウィンドウ部としてのIMDCT演算回路に置き換えた点のみである。そのため、固定小数点方式の逆変換計算装置においては上記順変換計算装置51と同様の問題が、浮動小数点方式の逆変換計算装置においては上記順変換計算装置61と同様の問題があった。

【0036】本発明は上記問題点を解決するためになされたものであって、第1の発明の目的は、回路規模を増大させることなく精度を高めることができる信号符号化装置を提供することにある。また、第2の発明の目的は、回路規模を増大させることなく精度を高めることができる信号復号化装置を提供することにある。また、第3の発明の目的は、第1の発明の信号符号化装置または第2の発明の信号復号化装置において、信号をデータとして処理する際に、オーバーフローを判定すると共に、オーバーフローしたデータを正確なデータに復元することにある。

【0037】

【課題を解決するための手段】第1の発明は、信号符号化装置において、入力信号波形を所定の周波数帯域毎にブロック化して分割すると共に、そのブロック内の信号を所定の時間毎にブロック化して切り出し、その分割されて切り出された各グループの信号を相互に独立な変換軸で変換して符号化する際に、各グループ毎に固定小数点方式を取り、その小数点位置を各グループ毎に任意に設定すると共に、変換のための演算処理において演算結果がオーバーフローした場合には、小数点位置を最適な値に設定し直すことをその要旨とする。

【0038】第2の発明は、信号復号化装置において、第1の発明の信号符号化装置から出力される各グループ毎の信号を相互に独立な変換軸で変換して復号化する際に、各グループ毎に固定小数点方式を取り、その小数点位置を各グループ毎に任意に設定すると共に、変換のための演算処理において演算結果がオーバーフローした場合には、小数点位置を最適な値に設定し直すことをその要旨とする。

【0039】第3の発明は、第1の発明の信号符号化装置または第2の発明の信号復号化装置において、前記各グループの信号をデータとして処理する際に、データのサインビットに適宜なビット数のマージンビットを付加して複数ビットから成るサインビットとし、前記演算処



理は当該マージンビットのビット数を越えないように設定して、前記オーバーフローは当該マージンビットの演算処理による変化から判定すると共に、そのマージンビットの変化からオーバーフローしたデータを正確なデータに復元することをその要旨とする。

#### 【 0 0 4 0 】

【作用】従って第 1 の発明によれば、各グループの信号に対して固定小数点方式による変換符号化がなされるが、その小数点位置は各グループ毎に任意に設定することができる。また、変換のための演算処理において演算結果がオーバーフローした場合には、小数点位置を最適な値に設定し直すことができる。そのため、入力信号波形の時間軸に対する変動が大きい場合や高調波成分が複雑に変動する場合においても、最適な小数点位置に基づく変換処理を行うことができ、高い変換精度を実現することができる。

【 0 0 4 1 】また、第 2 の発明によれば、各グループの信号に対して固定小数点方式による変換復号化がなされるが、その小数点位置は各グループ毎に任意に設定することができる。また、変換のための演算処理において演算結果がオーバーフローした場合には、小数点位置を最適な値に設定し直すことができる。そのため、第 1 の発明と同様に、入力信号波形の時間軸に対する変動が大きい場合や高調波成分が複雑に変動する場合においても、最適な小数点位置に基づく変換処理を行うことができ、高い変換精度を実現することができる。

【 0 0 4 2 】また、第 3 の発明によれば、前記オーバーフローは、サインビットに付加した適宜なビット数のマージンビットの演算処理による変化から判定することができる。さらに、そのマージンビットの変化からオーバーフローしたデータを正確なデータに復元することができる。

#### 【 0 0 4 3 】

【実施例】以下、本発明を具体化した一実施例を図 1 ～ 図 7 に従って説明する。尚、本実施例において、図 8 ～ 図 1 3 に示す従来例と同じ構成については符号を等しくしてその詳細な説明を省略する。

【 0 0 4 4 】図 1 に、本実施例のブロック回路図を示す。信号符号化装置 1 は、分割フィルタ 3 と各波形切り出し回路 2 と各順変換計算装置 4 と各量子化装置 5 とから構成されている。また、信号復号化装置 6 は、各逆量子化装置 7 と各逆変換計算装置 8 と各波形連結回路 1 0 と合成フィルタ 9 とから構成されている。

【 0 0 4 5 】QMF (Quadrature Mirror Filter) 等による分割フィルタ 3 は、入力信号を所定の周波数帯域毎にブロック化して分割する。各波形切り出し回路 2 は、分割フィルタ 3 にて周波数帯域毎にブロック化された時間軸上の入力信号を、所定の時間毎にブロック化して切り出す。そして、波形切り出し回路 2 にて所定の時間毎にブロック化された入力信号はそれぞれ、各順変換計算

装置 4 に入力される。

【 0 0 4 6 】この入力信号の時間軸上および周波数帯域毎のブロック化を、図 2 に従って説明する。図 2 に示す例では、分割フィルタ 3 により、入力信号は 3 つの周波数帯域  $0 \sim f_1$ ,  $f_1 \sim f_2$ ,  $f_2 \sim f_3$  にブロック化され、各ブロック A ～ C として分割される。そして、各波形切り出し回路 2 により、各ブロック A ～ C の入力信号は時間 T 1, T 2, T 3 … でブロック化され、各ブロック 1 ～ 3 として切り出される。これにより、各波形切り出し回路 2 からは、まず、ブロック 1 の各ブロック A ～ C であるグループ 1 A ～ 1 C に対応する入力信号が出力され、次に、ブロック 2 の各ブロック A ～ C であるグループ 2 A ～ 2 C に対応する入力信号が出力され、続いて、ブロック 3 の各ブロック A ～ C であるグループ 3 A ～ 3 C に対応する入力信号が出力される。

【 0 0 4 7 】ここで、順変換計算装置 4 は、分割フィルタ 3 による周波数帯域のブロック A ～ C 毎に設けられている（つまり、図 2 に示す例では、3 つの順変換計算装置が設けられている）。従って、各順変換計算装置 4 にはそれぞれ、まず、各グループ 1 A ～ 1 C に対応する入力信号が入力され、次に、各グループ 2 A ～ 2 C に対応する入力信号が入力され、続いて、各グループ 3 A ～ 3 C に対応する入力信号が入力される。

【 0 0 4 8 】各順変換計算装置 4 はそれぞれ、後記するように、時間軸上の入力信号を周波数軸上に変換する。各量子化装置 5 はそれぞれ、各順変換計算装置 4 の出力信号を量子化し、その量子化した信号を信号符号化装置 1 の出力信号（すなわち、入力信号を変換符号化した信号）として出力する。

【 0 0 4 9 】信号復号化装置 6 は、信号符号化装置 1 の周波数軸上の出力信号に信号符号化装置 1 とは逆の処理を施して、信号符号化装置 1 に入力された元の時間軸上の入力信号に復号化する。

【 0 0 5 0 】すなわち、各逆量子化装置 7 はそれぞれ、信号符号化装置 1 の出力信号である各量子化装置 5 の出力信号を逆量子化する。これにより、各逆量子化装置 7 の出力信号は、対応する各順変換計算装置 4 の出力信号と同じになる。

【 0 0 5 1 】各逆変換計算装置 8 はそれぞれ、後記するように、各逆量子化装置 7 の出力信号を周波数軸上から時間軸上に変換する。これにより、各逆変換計算装置 8 の出力信号は、対応する各波形切り出し回路 2 の出力信号と同じになる。

【 0 0 5 2 】各波形連結回路 1 0 は、各逆変換計算装置 8 の出力信号を順次連結する。これにより、各波形連結回路 1 0 の出力信号は、分割フィルタ 3 によって所定の周波数帯域毎にブロック化された時間軸上の入力信号と同じになる。

【 0 0 5 3 】QMF 等による合成フィルタ 9 は、各逆変換計算装置 8 の出力信号を合成する。これにより、合成

フィルタ 9 の出力信号 (すなわち、信号復号化装置 6 の出力信号) は、信号符号化装置 1 に入力された元の入力信号と同じになる。

【0054】図 3 に、順変換計算装置 4 のブロック回路図を示す。順変換計算装置 4 は、固定小数点演算を行う MDCT 演算回路 5 3 と RAM 2 1 と小数点位置レジスタ 2 2 とオーバーフロー検出回路 2 3 とから構成されている。

【0055】ここでは前記入力バッファとして働く RAM 2 1 は、前記グループ (1A~1C, 2A~2C, 3A~3C) の内の対応する 1 つのグループの入力信号を入力して保持する。この 1 つのグループの入力信号の実時間上のサンプル点は従来例と同様に N 個であるとする (つまり、時間軸上の N 点の離散値によって信号が表される)。

【0056】前記順変換ウィンドウ部および線形順変換部としての MDCT 演算回路 5 3 は、オーバーフロー検出回路 2 3 を介して RAM 2 1 から所望の入力信号を読み出しては前記した演算処理を行い、その結果を再び RAM 2 1 に書き込むという動作を繰り返し行う。

【0057】これにより、N サンプルの入力信号の全てについて所定の演算処理が終了すると、RAM 2 1 に保持されていた N サンプルの入力信号は、入力信号を周波数軸上の N 点の離散値に変換した信号である N サンプルの出力信号に置き代わることになる。すなわち、順変換計算装置 4 の入力信号は時間軸上から周波数軸上に直交変換される。

【0058】ここで、MDCT 演算回路 5 3 が固定小数点演算を行う点と、RAM 2 1 中のデータの表示方法が固定小数点表示である点については、図 8 に示す固定小数点方式をとる従来例と同じである。

【0059】但し、その固定小数点位置は小数点位置レジスタ 2 2 に格納されており、任意に設定することができる点が当該従来例と異なっている。また、オーバーフロー検出回路 2 3 を設けて RAM 2 1 から読み出したデータがオーバーフローしているかどうかを検出し、その検出結果に基づいて小数点位置レジスタ 2 2 に格納されている固定小数点位置を設定し直すことができる点も当該従来例と異なっている。

【0060】すなわち、MDCT 演算回路 5 3 においては、1 回の演算処理でデータの値が 2 倍を越えないように (つまり、演算処理によってデータが繰り上がった場合でも、その繰り上がりが 1 ビットを越えないように) アルゴリズムが設定されている。

【0061】そして、従来例では 1 ビットであったサインビットに 1 ビット分のマージンビットを加えて、2 ビットでサインビットが構成されている。これにより、データが正の最大値「1111…」の場合は、サインビット「00」がその先頭に付加されて「001111…」と表されることになる。一方、データが負の最大値「0

000…」の場合は、サインビット「11」がその先頭に付加されて「110000…」と表されることになる。

【0062】従って、正の最大値「001111…」であったデータが繰り上がった場合 (すなわち、オーバーフローした場合)、繰り上がりは 1 ビットであるため、そのデータは「011111…」になる。一方、負の最大値「110000…」であったデータが繰り上がった場合 (すなわち、オーバーフローした場合)、繰り上がりは 1 ビットであるため、そのデータは「100000…」になる。

【0063】このように、データがオーバーフローした場合、先頭のサインビットは変わらずにサインビットとしての機能を保ち、2 ビット目のサインビットは反転してデータがオーバーフローしたことを表す。

【0064】そこで、オーバーフロー検出回路 2 3 によってサインビットを検出することにより、RAM 2 1 から読み出したデータのオーバーフローの有無を検出することができる。そして、データがオーバーフローしていた場合、オーバーフロー検出回路 2 3 は小数点位置レジスタ 2 2 に格納されている固定小数点位置を 1 ビットシフトダウンして設定し直す。

【0065】そのオーバーフロー検出回路 2 3 の動作を、図 5 に従って説明する。図 5 に示す例では、N サンプルの全データ (すなわち、全信号) のデータ長を 24 ビットに設定し、23, 24 ビットをサインビットとして、残りの 22 ビットで 1 つのデータが表される。このとき、その固定小数点位置は小数点位置レジスタ 2 2 に格納されている。

【0066】図 5 の (a) に示すように、RAM 2 1 から読み出したデータが正の値でオーバーフローしていない場合、そのサインビット SS は「00」になっている。一方、RAM 2 1 から読み出したデータが負の値でオーバーフローしていない場合、そのサインビット SS は「11」になっている。オーバーフロー検出回路 2 3 は、そのサインビット SS から当該データがオーバーフローしていないことを検出しオーバーフローフラグ「0」を立てる。

【0067】このように RAM 2 1 から読み出したデータがオーバーフローしていない場合、MDCT 演算回路 5 3 は、小数点位置レジスタ 2 2 に格納されている固定小数点位置に基づいて固定小数点演算を行う。

【0068】このとき、RAM 2 1 から読み出したデータが正の最大値「001111…」であった場合、MDCT 演算回路 5 3 における演算処理によりデータは繰り上がってオーバーフローするが、その繰り上がりは 1 ビットになるように MDCT 演算回路 5 3 のアルゴリズムは設定されている。そのため、MDCT 演算回路 5 3 から RAM 2 1 に書き込まれるデータは「011111…」になる。同様に、RAM 2 1 から読み出したデータ

が負の最大値「110000…」であった場合、MDCT演算回路53における演算処理によりデータは繰り上がってオーバーフローするが、その繰り上がりは1ビットになる。そのため、MDCT演算回路53からRAM21に書き込まれるデータは「100000…」になる。

【0069】すなわち、MDCT演算回路53における演算処理によりデータがオーバーフローした場合、そのサインビットSXは「01」または「10」になる。ここで、先頭のサインビットSはサインビットとしての機能を保ち、2ビット目のサインビットXはデータがオーバーフローしたことを表している。

【0070】図5の(b)に示すように、これらオーバーフローしたデータがRAM21から読み出されると、オーバーフロー検出回路23はサインビットSXから当該データがオーバーフローしていることを検出しオーバーフローフラグ「1」を立てる。

【0071】次に、オーバーフロー検出回路23は、小数点位置レジスタ22に格納されている固定小数点位置を1ビット分だけ右にシフト（すなわち、シフトダウン）させて正規化し直し、新しい固定小数点位置を設定する。また、オーバーフロー検出回路23は、2ビット目のサインビットを先頭のサインビットと同じにする。すなわち、サインビットSXが「01」の場合は「00」にし、「10」の場合は「11」にする。これにより、オーバーフローしたデータを正確なデータに復元することができる。

【0072】そして、MDCT演算回路53は、オーバーフロー検出回路23によって設定し直され小数点位置レジスタ22に格納された新しい固定小数点位置に基づいて、復元したデータの固定小数点演算を行う。

【0073】以後は、データがオーバーフローする度に同様の処理を行う。従って、MDCT演算回路53による演算処理においてデータがオーバーフローする度に、小数点位置レジスタ22に格納されている固定小数点位置が1ビットシフトダウンした値に設定し直される。

【0074】このように、順変換計算装置4は、1つのグループの入力信号に対して任意の固定小数点位置による演算処理が終了したら、次のグループの入力信号に対しては前のグループとは別の固定小数点位置を設定して演算処理を行うことができる。また、複数設けられている順変換計算装置4のそれぞれが、別の固定小数点位置による演算処理を行うこともできる。そして、MDCT演算回路53による演算処理においてデータがオーバーフローしても、そのデータをRAM21から読み出した時点で正確に復元することができる。さらに、データがオーバーフローする度に、小数点位置レジスタ22に格納されている固定小数点位置を設定し直すため、1つのグループの入力信号に対して固定小数点位置を常に最適に保つことができる。

【0075】図4に、逆変換計算装置8のブロック回路図を示す。順変換計算装置4と逆変換計算装置8との違いは、MDCT演算回路53をIMDCT演算回路31に置き換えた点のみである。すなわち、逆変換計算装置8は、固定小数点演算を行うIMDCT演算回路31とRAM21と小数点位置レジスタ22とオーバーフロー検出回路23とから構成されている。

【0076】ここでは前記出力バッファとして働くRAM21は、量子化装置5の出力信号（すなわち、分割フィルタ3から出力された1つのグループの〔Nサンプルの〕入力信号を周波数軸上に変換した後に量子化した信号）を保持する。

【0077】前記線形逆変換部および逆変換ウィンドウ部としてのIMDCT演算回路31は、RAM21から所望の信号を読み出しては前記した演算処理を行い、その結果を再びRAM21に書き込むという動作を繰り返す。

【0078】これにより、RAM21のNサンプルの信号の全てについて所定の演算処理が終了すると、RAM21に保持されていた周波数軸上のNサンプルの信号は、その信号を時間軸上のN点の離散値に変換（ここでは、逆変換）した信号（すなわち、順変換計算装置1の入力信号）であるNサンプルの出力信号に置き代わることになる。すなわち、逆変換計算装置8の入力信号は周波数軸上から時間軸上に直交変換される。

【0079】ここで、RAM21中のデータの表示方法は順変換計算装置4と同様に固定小数点表示であり、IMDCT演算回路31は固定小数点演算を行う。また、当該固定小数点位置は小数点位置レジスタ22に格納されており、任意に設定することができる。さらに、オーバーフロー検出回路23を設けてRAM21から読み出したデータがオーバーフローしているかどうかを検出し、その検出結果に基づいて小数点位置レジスタ22に格納されている固定小数点位置を設定し直すことができる。

【0080】尚、IMDCT演算回路53はMDCT演算回路53と同様に、1回の演算処理でデータの値が2倍を越えないようにアルゴリズムが設定されている。また、逆変換計算装置8においても順変換計算装置4と同様に、2ビットでサインビットが構成されている。逆変換計算装置8のオーバーフロー検出回路23の動作は、順変換計算装置4のそれと同じであるためここでは説明を省略する。

【0081】このように、逆変換計算装置8は順変換計算装置4と同様に、1つのグループの入力信号（すなわち、量子化装置5の出力信号）に対して任意の固定小数点位置による演算処理が終了したら、次のグループの入力信号に対しては前のグループとは別の固定小数点位置を設定して演算処理を行うことができる。また、複数設けられている逆変換計算装置8のそれぞれが、別の固定

小数点位置による演算処理を行うこともできる。そして、IMDCT演算回路31による演算処理においてデータがオーバーフローしても、そのデータをRAM21から読み出した時点で正確に復元することができる。さらに、データがオーバーフローする度に、小数点位置レジスタ22に格納されている固定小数点位置を設定し直すため、1つのグループの入力信号に対して固定小数点位置を常に最適に保つことができる。

【0082】このように、本実施例の順変換計算装置4および逆変換計算装置8は、時間軸上の入力信号を所定の時間毎にブロック化した各ブロック（図2に示す、ブロック1～3）の1つ1つの処理については固定小数点方式をとっている。その一方で、順変換計算装置4および逆変換計算装置8は、当該時間軸上の各ブロック毎にそれぞれ別の固定小数点位置を設定することができる。つまり、順変換計算装置4および逆変換計算装置8は、各データ単位ではなく、時間軸上の各ブロック単位で浮動小数点方式をとっていることになる。

【0083】また、順変換計算装置4および逆変換計算装置8は、時間軸上の各ブロックを所定の周波数帯域毎にブロック化した各ブロック（図2に示す、ブロックA～C）毎に設けられている。そのため、本実施例では、当該周波数帯域毎の各ブロックの1つ1つの処理については固定小数点方式をとっている。その一方で、各順変換計算装置4および各逆変換計算装置8はそれぞれ別の固定小数点位置を設定することができるため、周波数帯域毎の各ブロック毎にそれぞれ別の固定小数点位置を設定することができる。つまり、本実施例では、周波数帯域毎の各ブロック毎に順変換計算装置4および逆変換計算装置8を設けることにより、各データ単位ではなく、周波数帯域毎の各ブロック単位で浮動小数点方式をとっていることになる。

【0084】すなわち、本実施例では、前記各グループ（1A～1C、2A～2C、3A～3C）単位で浮動小数点方式をとっていることになる。従って、各グループ毎に最適な小数点位置を設定することができる。

【0085】図6および図7は、Nサンプルの全データ（すなわち、全信号）のデータ長を24ビットに設定し、時間軸上および周波数帯域毎の各ブロックを図2に示すように設定した場合における、各順変換計算装置4または各逆変換計算装置8の各RAM21内の信号の格納状態を模式的に示したものである。尚、図2に示す例では、順変換計算装置および逆変換計算装置8がそれぞれ3つずつ設けられているため、RAM21もそれぞれ3つずつ設けられていることになる。

【0086】図7に示すように、入力信号のレベルが全周波数帯域にわたって小さくときには、各RAM21の小数点位置を上位のビット側に設定することにより、S/N比を大きくすることができる。また、図6に示すように、入力信号のレベルが全周波数帯域にわたって大き

なときには、各RAM21の小数点位置を図7に比べて下位のビット側に設定することにより、やはりS/N比を大きくすることができる。

【0087】このように本実施例においては、時間軸上の入力信号を所定の時間毎にブロック化し、その時間軸上の各ブロックを所定の周波数帯域毎にブロック化して各グループとしている。そして、周波数帯域毎の各ブロック毎に順変換計算装置4および逆変換計算装置8を設けている。さらに、各順変換計算装置4および各逆変換計算装置8は固定小数点方式をとっているが、その固定小数点位置は時間軸上の各ブロック毎に任意に設定することができる。そのため、本実施例では、時間軸上の各ブロック単位で浮動小数点方式をとると共に、周波数帯域毎の各ブロック単位で浮動小数点方式をとっていることになる。すなわち、時間軸上の各ブロックを周波数帯域毎に分割した各グループ単位で浮動小数点方式をとっていることになる。

【0088】従って、本実施例では、従来例の固定小数点方式に比べて、レベル差が大きな信号を扱う場合でも十分なS/N比を得ることができ、精度を高めることができる。

【0089】また、MDCT演算回路53およびIMDCT演算回路31においては1回の演算処理でデータの値が2倍を越えないようにアルゴリズムを設定し、サインビットに1ビット分のマージンビットを加えて2ビットでサインビットを構成している。そして、オーバーフロー検出回路23により、RAM21から読み出したデータのサインビットから当該データがオーバーフローしているかどうかを検出する。データがオーバーフローしている場合、オーバーフロー検出回路23は、小数点位置レジスタ22に格納されている固定小数点位置を設定し直す。また、オーバーフロー検出回路23は、サインビットに基づいて当該オーバーフローしたデータを正確に復元してMDCT演算回路53またはIMDCT演算回路31に転送している。

【0090】従って、本実施例では、従来例の固定小数点方式のように、データが一旦オーバーフローすると、それ以降の演算結果は全てオーバーフローすることになり意味をなさなくなるということはない。

【0091】また、各順変換計算装置4および各逆変換計算装置8には、それぞれ1つの小数点位置レジスタ22しか設けていない。そのため、本実施例では、図9に示すように各データ毎に小数点位置レジスタ64を設けている従来例の浮動小数点方式に比べて、回路規模を小さくすることができる。

【0092】さらに、MDCT演算回路53およびIMDCT演算回路31はそれぞれ固定小数点演算を行っているため前記スケール合わせを行う必要がなく、従来例の浮動小数点方式に比べて回路規模を小さくすることができる。

【0093】ところで、本実施例はオーディオ信号や画像信号の処理に使用することができる。例えば、オーディオ信号の処理に用いる場合には、周波数帯域毎の各グループA～Cをそれぞれ、低域、中域、高域に対応させることが考えられる。

【0094】すなわち、一般の音楽ソースのエネルギーレベルは、低域、中域、高域がそれぞれ6 : 3 : 1程度の割合になっている。そこで、従来の固定小数点方式をオーディオ信号の処理に利用する場合には、各帯域のエネルギーレベルに合わせて、設計時に小数点位置を設定

10 していた。

【0095】ところが、トライアングルやハイハット等の打楽器を多用した音楽ソースでは高域のレベルだけが極端に高くなることがある。一方、パイプオルガンやバスドラムが入っている音楽ソースでは低域のレベルだけが極端に高くなることがある。このように各帯域でレベルが変動しても、本実施例によれば、そのレベル変動に合わせて各帯域毎に小数点位置を設定し直すことができるため、高い精度で変換符号化または復号化することができる。従って、本実施例をオーディオ信号の処理に利用すれば、レベル変動に関係なく聴感に即して各帯域を

バランス良く明瞭に記録・再生することができる。

【0096】尚、本発明は上記実施例に限定されるものではなく、以下のように実施してもよい。

1) 量子化装置5および逆量子化装置7を省き、順変換計算装置4の出力信号を直接、逆変換計算装置8に入力させるようにする。

【0097】2) 信号符号化装置1の出力信号を直接、信号復号化装置2に入力させるのではなく、信号符号化装置1の出力信号を一旦記録媒体(磁気テープ、磁気ディスク、光ディスク、光磁気ディスク等)に記録させ、記録媒体から読み出した信号を信号復号化装置2に入力させるようにする。

【0098】3) 周波数帯域毎のブロック化を図2に示すような3つではなく、2つ又は4つ以上にして行う。この周波数帯域毎のブロック数が多いほど精度は向上するが、その反面、回路規模は増大する。

【0099】4) MDCT演算回路53およびIMDCT演算回路31において、1回の演算処理でデータの値がm倍(mは2以上の整数)を越えないようにアルゴリズムを設定する。この場合は、サインビットに(m-1)ビット分のマージンビットを加えてmビットでサインビットを構成し、上記実施例と同様に具体化する。

【0100】5) MDCTでなく、アダマール変換、カルネシレーブ変換、ルジャンドル変換、DFT、DCT等の他の直交変換によって具体化する。

6) 分割フィルタ3によって所定の周波数帯域毎にブロック化された入力信号を、各波形切り出し回路2によって所定の時間毎にブロック化して切り出すのではなく、分割フィルタ3と各波形切り出し回路2の処理とを逆に

する。

【0101】すなわち、時間軸上の入力信号を先ず1つの波形切り出し回路2によって所定の時間毎にブロック化して切り出した後、分割フィルタ3によって所定の周波数帯域毎にブロック化して分割する。

【0102】

【発明の効果】以上詳述したように、第1の発明によれば、回路規模を増大させることなく精度を高めることが可能な信号符号化装置を提供することができるという優れた効果がある。また、第2の発明によれば、回路規模を増大させることなく精度を高めることが可能な信号復号化装置を提供することができるという優れた効果がある。また、第3の発明によれば、第1の発明の信号符号化装置または第2の発明の信号復号化装置において、信号をデータとして処理する際に、オーバーフローを判定すると共に、オーバーフローしたデータを正確なデータに復元することができるという優れた効果がある。

【図面の簡単な説明】

【図1】本発明を具体化した一実施例のブロック回路図である。

【図2】入力信号の時間軸上および周波数帯域毎のブロック化の一例を説明するための説明図である。

【図3】一実施例の順変換計算装置4のブロック回路図である。

【図4】一実施例の逆変換計算装置8のブロック回路図である。

【図5】オーバーフロー検出回路23の動作を説明するための説明図である。

【図6】一実施例の各RAM21内の信号の格納状態を説明するための模式図である。

【図7】一実施例の各RAM21内の信号の格納状態を説明するための模式図である。

【図8】固定小数点方式をとる従来例の順変換計算装置51のブロック回路図である。

【図9】浮動小数点方式をとる従来例の順変換計算装置61のブロック回路図である。

【図10】固定小数点表示のRAM内におけるデータの格納状態を示す説明図である。

【図11】浮動小数点表示のRAM内におけるデータの格納状態を示す説明図である。

【図12】固定小数点方式をとる従来例のRAM52内の信号の格納状態を説明するための模式図である。

【図13】固定小数点方式をとる従来例のRAM52内の信号の格納状態を説明するための模式図である。

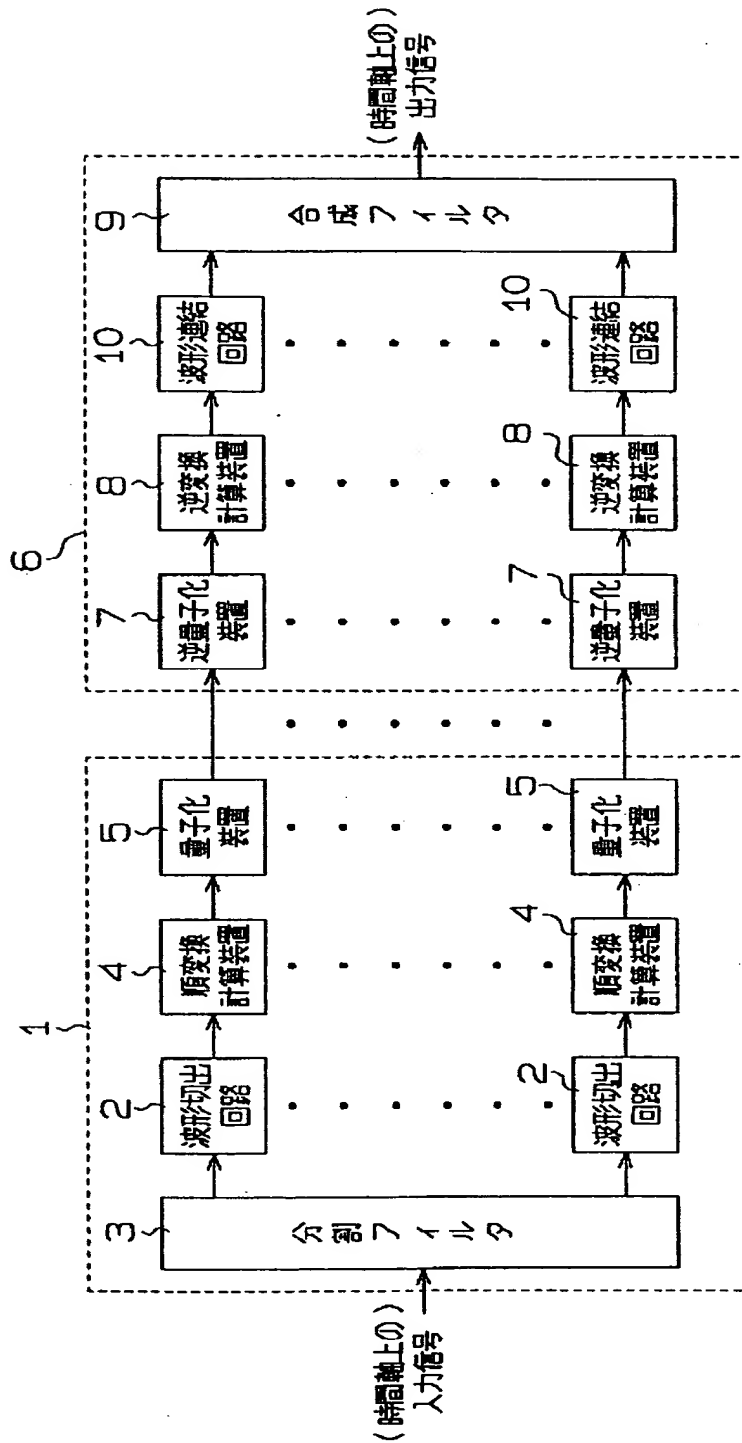
【符号の説明】

- 1 信号符号化装置
- 2 波形切り出し回路
- 3 分割フィルタ
- 4 順変換計算装置
- 5 量子化装置

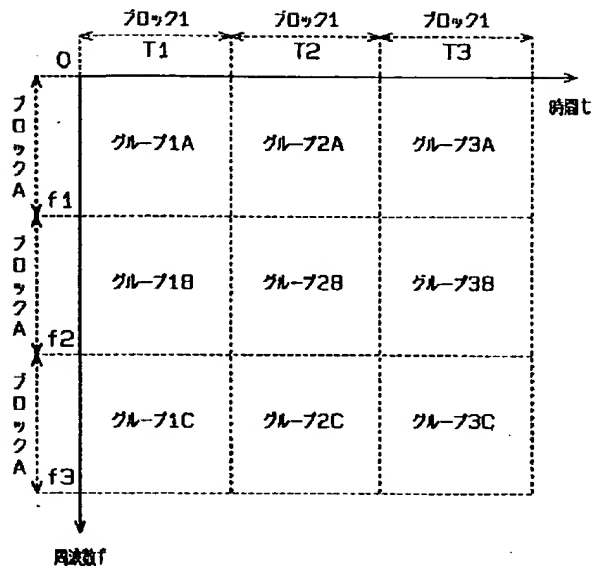
- 6 信号復号化装置
- 7 逆量子化装置
- 8 逆変換計算装置
- 9 合成フィルタ

- 10 波形連結回路
- 22 小数点位置レジスタ
- 23 オーバーフロー検出回路

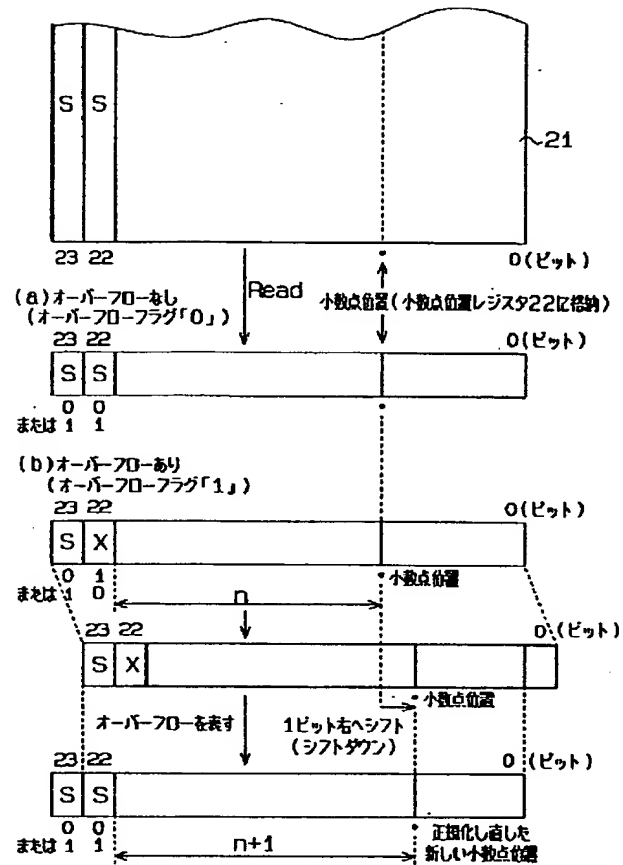
【図1】



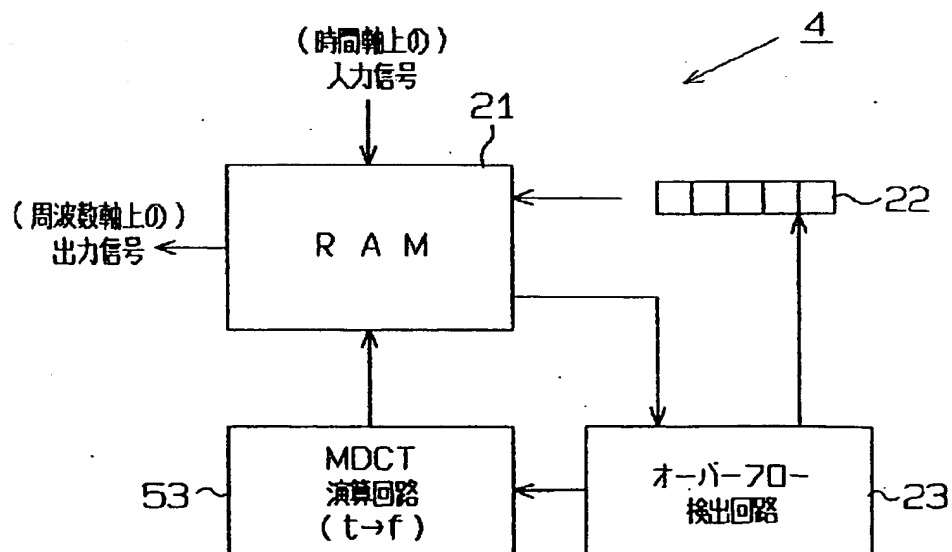
【図 2】



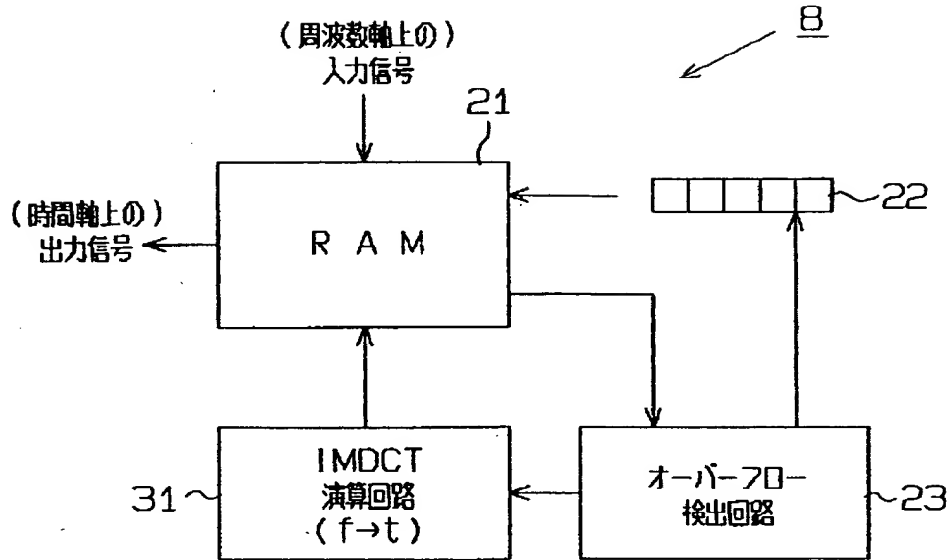
【図 5】



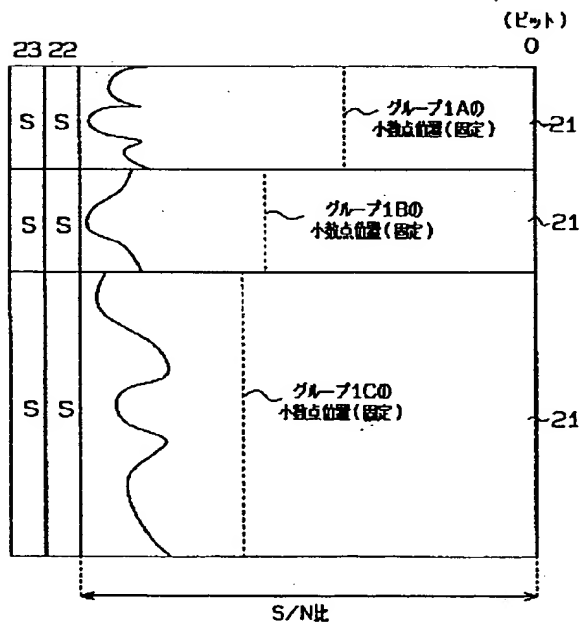
【図 3】



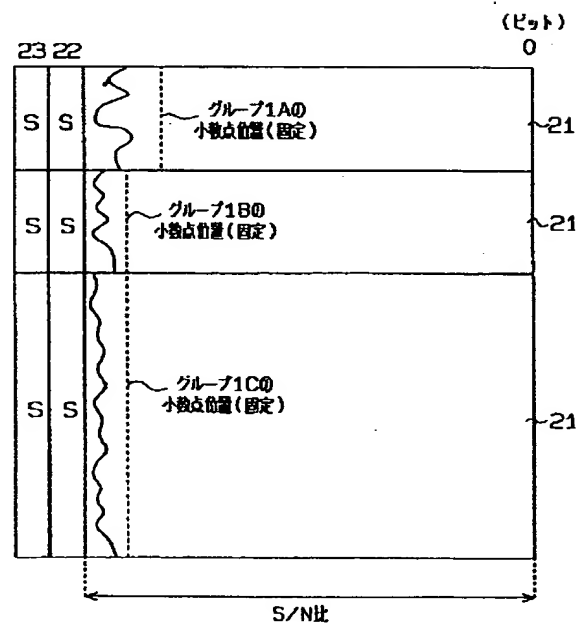
【図4】



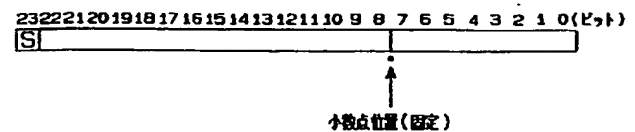
【図6】



【図7】

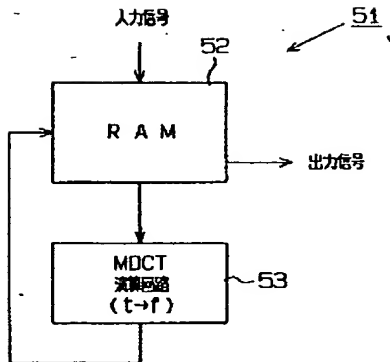


【図10】

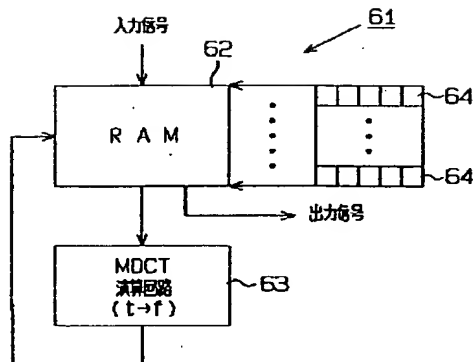




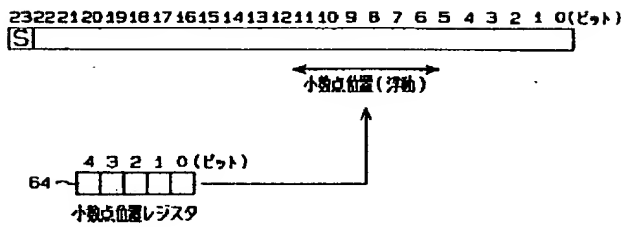
【図8】



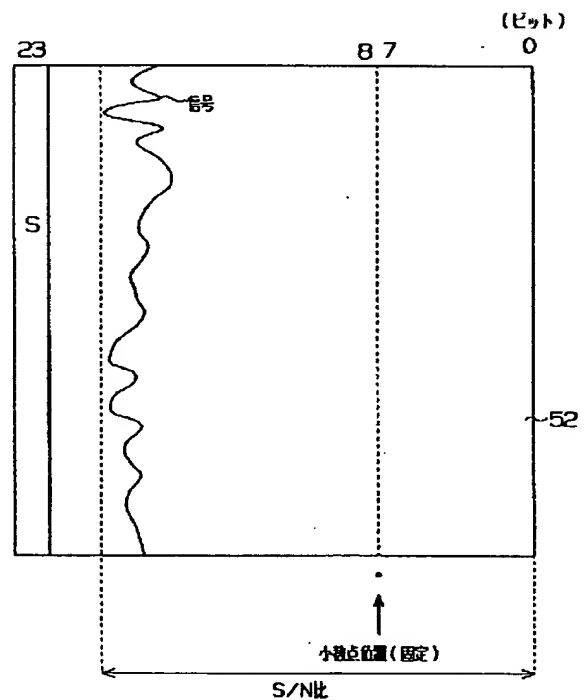
【図9】



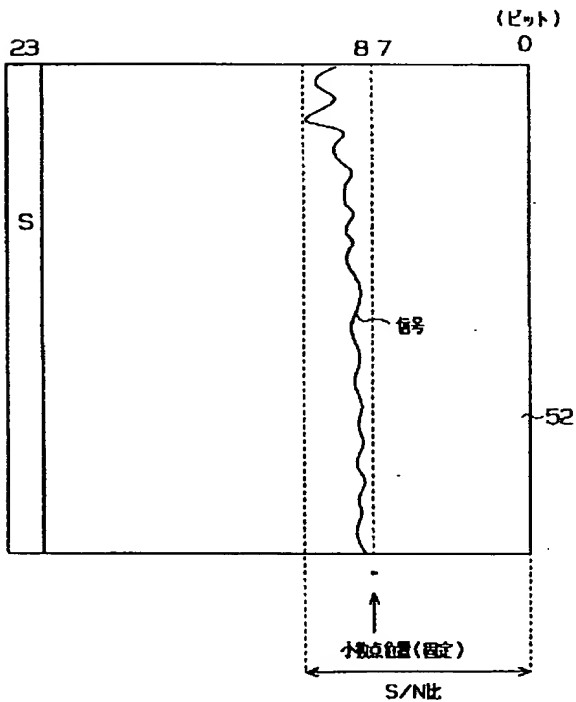
【図11】



【図12】



【図13】



*This Page Blank (uspto)*